



# Programozási nyelvek

## 4. előadás



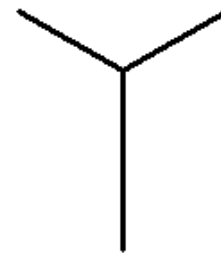


# Logo – fa variációk



## Fa rajzolása rekurzívan

A fa egy törzsből áll, amelynek tetején két ág nő ki, s mindkettő tulajdonképpen egy-egy alacsonyabb, rövidebb törzsű fa. Az ábrában önmaga felfedezhető!



eljárás fa :év :hossz

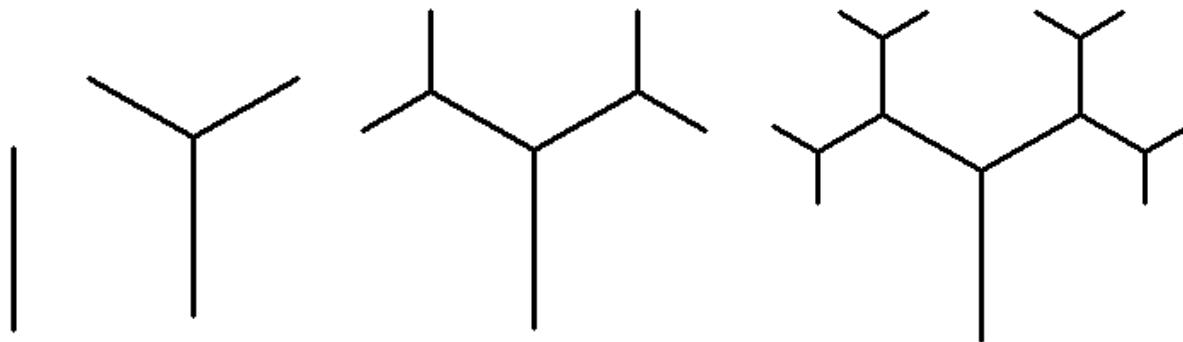
előre :hossz

ha :év>1 [balra 60 fa :év-1 :hossz/2 jobbra 120

fa :év-1 :hossz/2 balra 60]

hátra :hossz

vége



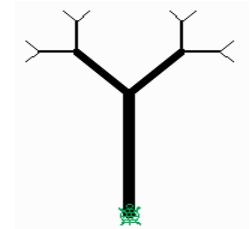


# Logo – fa variációk



## Fa rajzolása rekurzívan

A fa egy törzsből áll, amelynek tetején két ág nő ki, az ágak nemcsak rövidülnek, hanem vékonyodnak is.



```
eljárás fa :év :hossz :vast
  tollvastagság! :vast előre :hossz
  ha :év>1 [balra 60 fa :év-1 :hossz/2 :vast/2
            jobbra 120 fa :év-1 :hossz/2 :vast/2
            balra 60]
  tollvastagság! :vast hátra :hossz
vége
```

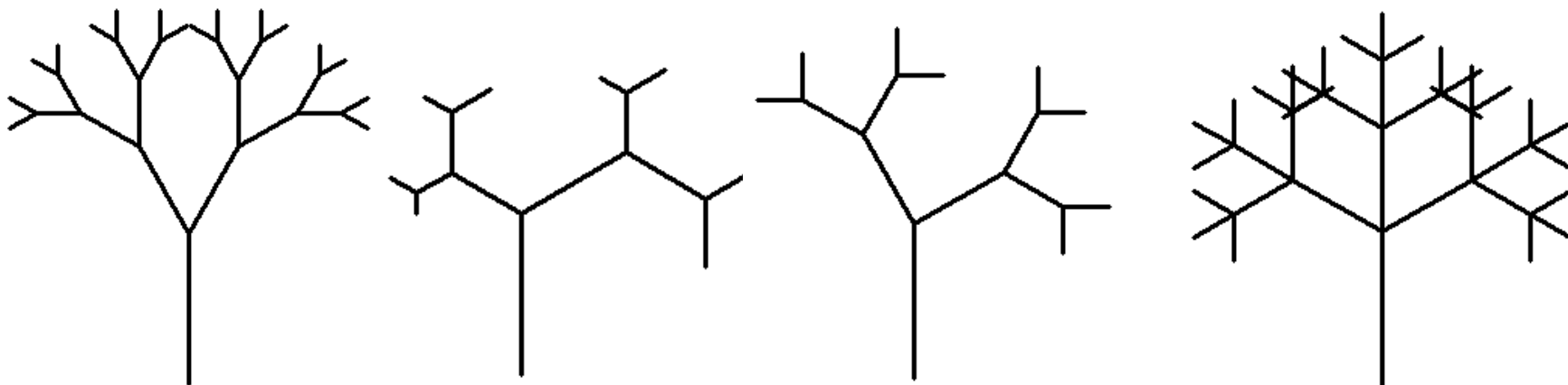




# Logo – fa variációk



## Fa variációk



```
eljárás fa :év :hossz
```

```
  előre :hossz
```

```
  ha :év>1 [balra 30 fa :év-1 :hossz/2 jobbra 60  
            fa :év-1 :hossz/2 balra 30]
```

```
  hátra :hossz
```

```
vége
```





# Logo – fa variációk



## Fa variációk

A fa egy törzsből áll, amelynek tetején két ág nő ki, a legutolsó ágak zöldek, a többiek barnák.

```
eljárás fa :év :hossz
```

```
  hak :év=1 [tollszín! 2][tollszín! [90 45 0]]
```

```
  előre :hossz
```

```
  ha :év>1 [balra 60
```

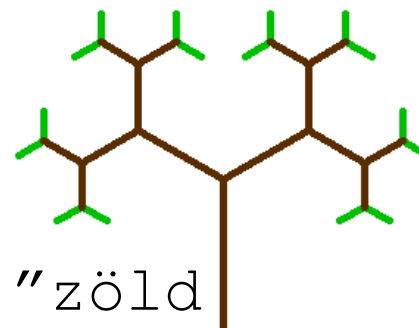
```
    fa :év-1 :hossz*2/3 jobbra 120
```

```
    fa :év-1 :hossz*2/3 balra 60]
```

```
  hak :év=1 [tollszín! 2][tollszín! [90 45 0]]
```

```
  hátra :hossz
```

```
vége
```



Lehetne: tollszín! "zöld"





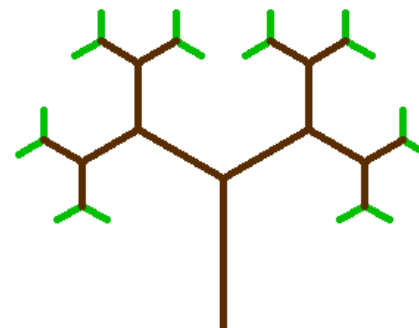
# Logo – fa variációk



```
eljárás fa :év :hossz
  hak :év=1 [tollszín! "zöld][tollszín! barna]
  előre :hossz
  ha :év>1 [balra 60
            fa :év-1 :hossz*2/3 jobbra 120
            fa :év-1 :hossz*2/3 balra 60]
  hak :év=1 [tollszín! 2][tollszín! [90 45 0]]
  hátra :hossz
vége
```

Probléma: "barna színnév nincs, helyette függvényt írunk:

```
eljárás barna
  eredmény [90 45 0]
vége
```





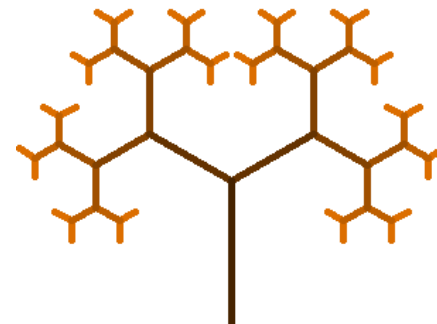
# Logo – fa variációk



## Fa variációk

A fa egy törzsből áll, amelynek tetején két ág nő ki, az ágak a törzstől távolodva egyre világosabb barnák.

```
eljárás fa :év :hossz :szín
  tollszín! :szín előre :hossz
  ha :év>1 [balra 60
    fa :év-1 :hossz*2/3 világosít :szín
    jobbra 120
    fa :év-1 :hossz*2/3 világosít :szín
    balra 60]
  tollszín! :szín hátra :hossz
vége
```





# Logo – fa variációk

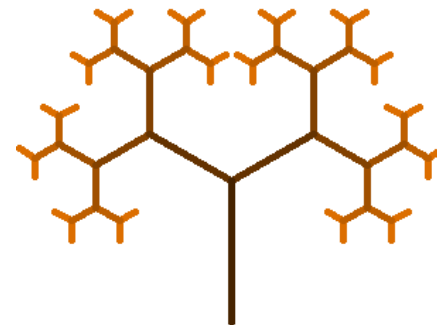


## Fa variációk

Az RGB szín 3 színkódból áll, ezt fel kell bontani 3 összetevőre, az összetevőket módosítani kell, majd újra összerakni RGB-kóddá! A kezdőszín: RGB[100 50 50] .

```
eljárás világosít :szín  
eredmény (lista 20+első :szín  
10+első elsőnélküli :szín  
utolsó :szín)
```

vége







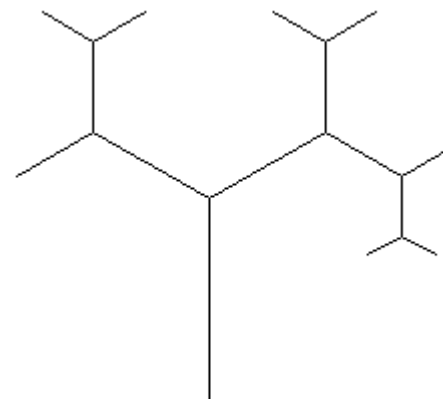
# Logo – fa variációk



## Fa variációk

A fa egy törzsből áll, amelynek tetején két ág nő ki, a baloldali kettő, a jobboldali egy szinttel kisebb.

```
eljárás fa :év :hossz
  előre :hossz
  ha :év>1 [balra 60
             fa :év-2 :hossz*2/3 jobbra 120
             fa :év-1 :hossz*2/3 balra 60]
  hátra :hossz
vége
```





# Logo – fa variációk



## Fa variációk

A fa egy törzsből és négy ágból áll (a törzsnek kinéző egyik rész is ág, de akár lehetne csak ágakból álló).

```
eljárás fa :év :hossz
```

```
  hak :év=1 [előre :hossz hátra :hossz]
```

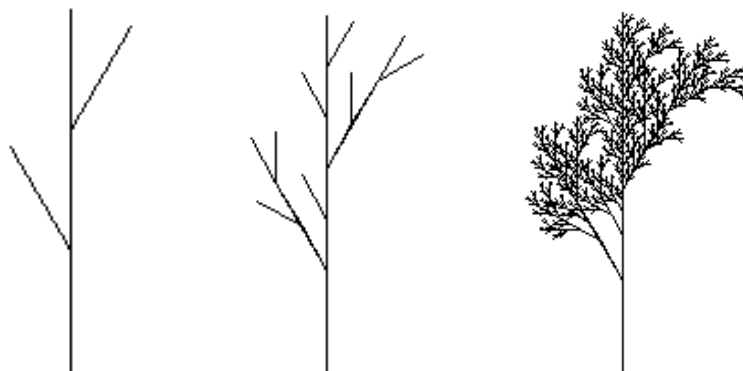
```
  [előre :hossz/2 balra 30 fa :év-1 :hossz/2
```

```
   jobbra 30 fa :év-1 :hossz/2 előre :hossz/2
```

```
   jobbra 30 fa :év-1 :hossz/2 balra 30
```

```
   fa :év-1 :hossz/2 hátra :hossz]
```

```
vége
```





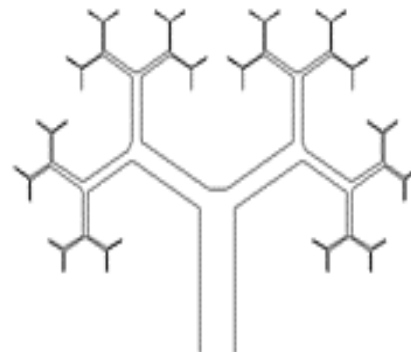
# Logo – fa variációk



## Fa variációk

A fa törzse egyik oldalát előrefelé, a másikat hátrafelé rajzoljuk, az ágak egyenlő szárú trapézhoz illeszkednek.

```
eljárás fa :év :hossz :vast
  előre :hossz
  hak :év=1 [jobbra 30 előre :vast jobbra 120
             előre :vast jobbra 30]
        [balra 60 fa :év-1 :hossz*2/3 :vast/2
          balra 30 előre :vast/2 balra 30
            fa :év-1 :hossz*2/3 :vast/2 balra 60]
  előre :hossz
vége
```





# Logo – fa variációk



## Fa variációk

A fa ágai lefelé hajlanak. A teknőc aktuális irányától függően az ágak vagy egyenes szakaszok, vagy bal-, illetve jobb-körívek.

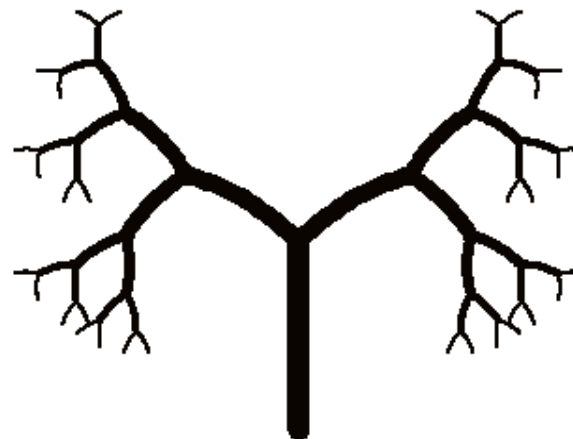
eljárás faág :év :hossz

```
ha :év>1 [balra 60 fa :év-1 :hossz*2/3
```

```
jobbra 120 fa :év-1 :hossz*2/3
```

```
balra 60]
```

vége





# Logo – fa variációk



## Fa variációk

Használjuk a teknőc irány érzékelőjét!

eljárás fa :év :hossz

  tollvastagság! :év\*2

  hak vagy irány=0 irány=180

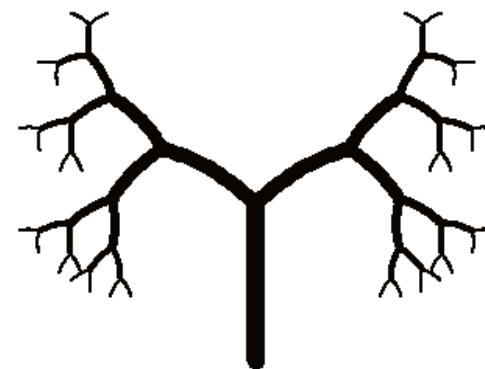
    [előre :hossz faág :év :hossz hátra :hossz]

  [hak irány<180

    [jobbív :hossz faág :év :hossz jobbív -:hossz]

    [balív :hossz faág :év :hossz balív -:hossz]

vége





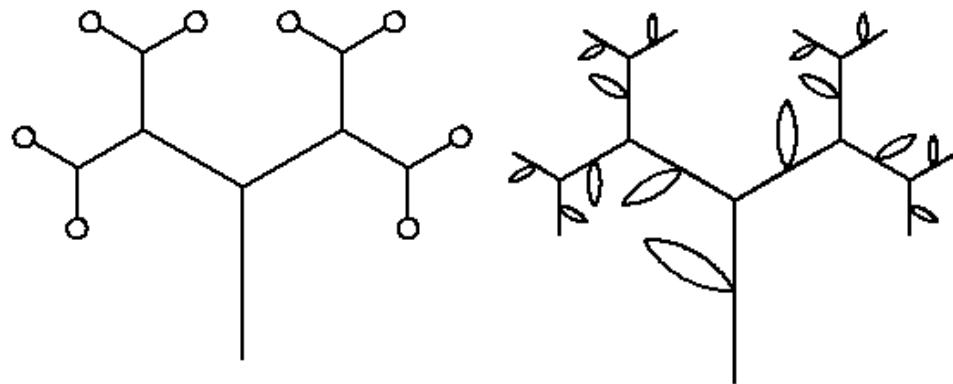
# Logo – fa variációk



## Fa variációk

Az első változatban az ágak végén virágok nőnek, a másodikban az ágak közepén levelek. Az első megoldása:

```
eljárás fa :év :hossz
  előre :hossz
  hak :év>1 [balra 45
             fa :év-1 :hossz/2 jobbra 90
             fa :év-1 :hossz/2 balra 45] [virág]
  hátra :hossz
vége
```





# Logo – Fraktál



## Fraktál – Cantor halmaz

0. lépés: vegyünk egy szakaszt

1. lépés: vágjuk ki a középső egyharmadát

2. lépés: a keletkezett két szakasznak is vágjuk ki a középső egyharmadát

3. lépés: a keletkezett összes szakasznak vágjuk ki a középső egyharmadát

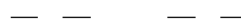
Amit végtelen lépés után kapnánk, az a Cantor halmaz, vagy Cantor por.



0. lépés



1. lépés



2. lépés



3. lépés



# Logo – Fraktál



## Fraktál – Cantor halmaz

eljárás cantor :lépés :hossz

hak :n=0 [előre :hossz]

[cantor :lépés-1 :hossz/3

tollatfel előre :hossz/3 tollatle

cantor :lépés-1 :hossz/3]

vége







# Logo – Fraktál

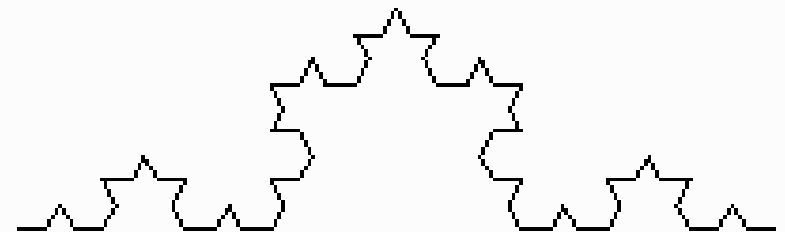
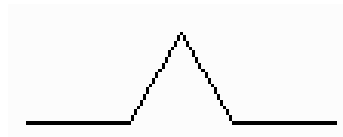


## Fraktál - Koch görbe

Egyik nevezetes fraktál az ún. Koch görbe. Ez egy rekurzív definícióval adott görbe, amelynek adott szintű változatai rajzolhatók meg.

```
eljárás Koch :szint :hossz
  hak :szint = 1 [előre :hossz]
  [Koch :szint-1 :hossz/3 balra 60
   Koch :szint-1 :hossz/3 jobbra 120
   Koch :szint-1 :hossz/3 balra 60
   Koch :szint-1 :hossz/3]
```

vége





# Logo – Fraktál

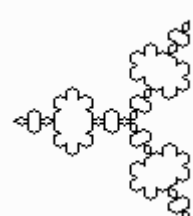
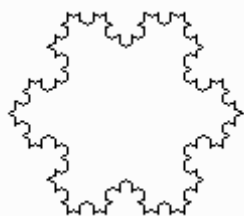


## Fraktál - kristály és hópehely

A Koch-görbe nagyon szép hópehelyszerű, illetve kristályszerű ábrákat adhat, ha a kiinduló egyenes szakaszt háromszöggel helyettesítjük. Másféle ábrát kapunk, ha a háromszöghöz képest kifelé, illetve ha befelé rajzoljuk a Koch görbét.

```
eljárás hópehely :szint :hossz  
  ismétlés 3 [Koch :szint :hossz jobbra 120]  
vége
```

```
eljárás kristály :szint :hossz  
  ismétlés 3 [Koch :szint :hossz balra 120]  
vége
```



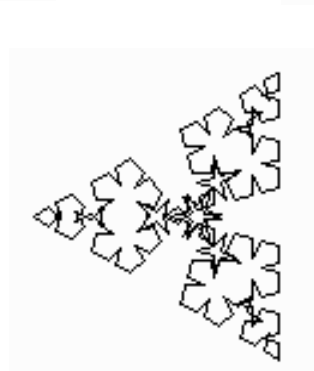
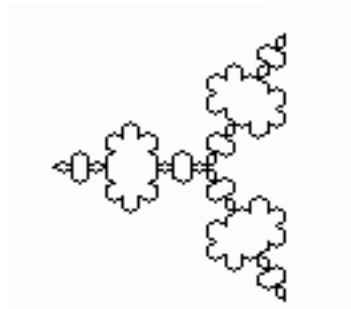
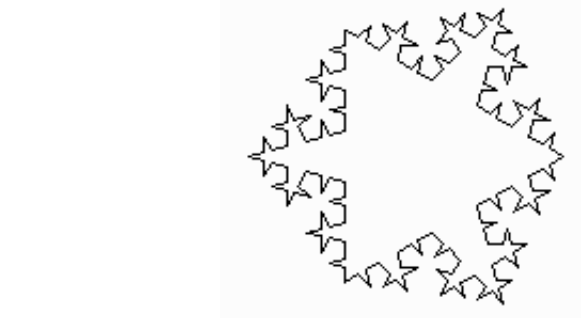
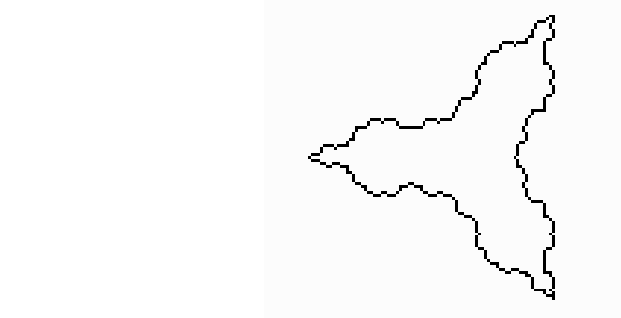
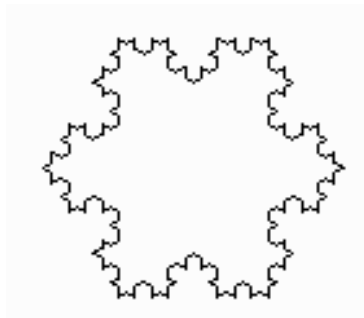


# Logo – Fraktál



## Fraktál - kristály és hópehely

Az eredeti Koch-görbében a balra-, illetve jobbrafordulás szögét változtatva különlegesen szép ábrákat kaphatunk.





# Logo – Fraktál



## Fraktál - területkitöltés

Ha a fordulatok szöge 90, illetve 180 fok, akkor egy háromszög alakú területet lefedő négyzethálót kapunk, s elég magas szintű ábrát rajzolva befesthetjük az egész háromszöget.

eljárás Koch :szint :hossz

hak :szint = 1

[előre :hossz]

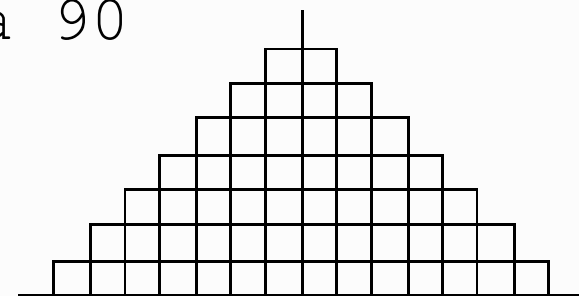
[Koch :szint-1 :hossz/2 balra 90

Koch :szint-1 :hossz/2 jobbra 180

Koch :szint-1 :hossz/2 balra 90

Koch :szint-1 :hossz/2]

vége





# Logo – Fraktál

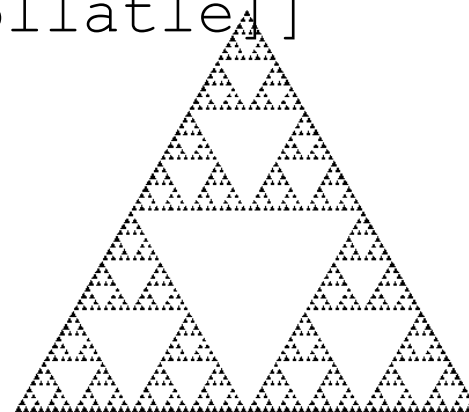
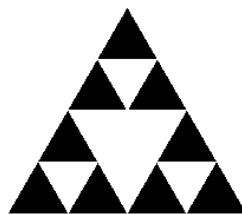
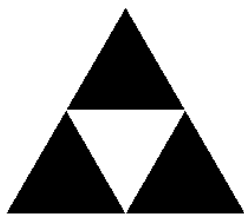


## Fraktál – Sierpinski háromszög

A Sierpinski háromszög olyan fraktál, melynek előállításakor szakasz helyett egy egyszerű síkidomból, egy szabályos háromszögből kell kiindulnunk.

```
eljárás sierp :lépés :hossz
  ha :lépés>0 [kivág :hossz
    ismétlés 3 [sierp :lépés-1 :hossz/2 tollatfel
      előre :hossz jobbra 120 tollatle]]]
```

vége





# Logo – Fraktál



## Fraktál – Sierpinski háromszög

eljárás kivág :h

tollatfel jobbra 90 előre :h balra 120 tollatle

háromszög :h 15

tollatfel jobbra 120 hátra :h balra 90

vége

eljárás háromszög :h :szín

ismétlés 3 [előre :h jobbra 120]

töltőszín! :szín tollatfel jobbra 30 előre 3

tölt hátra 3 balra 30 tollatle

vége



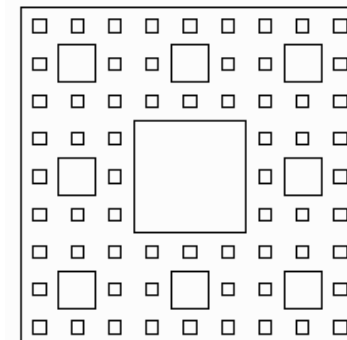
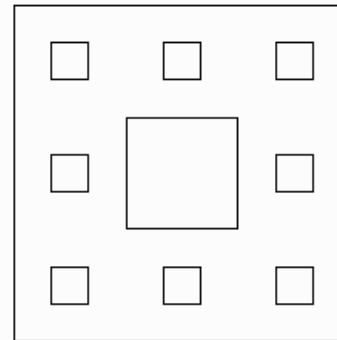
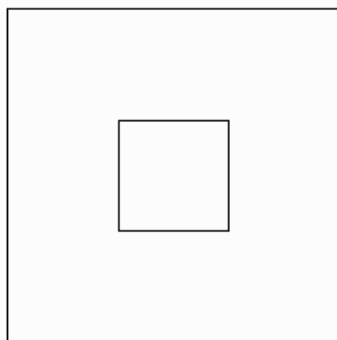


# Logo – Fraktál

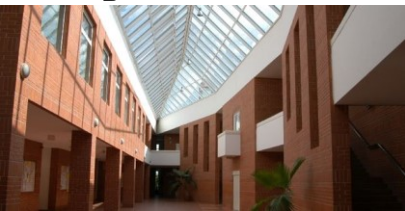


## Fraktál – Sierpinski csipke

Egy másik érdekes, síkbeli alakzat a Sierpinski csipke, amely egy négyzetlapból keletkezik úgy, hogy újabb és újabb belső négyzeteket vágunk ki.



```
eljárás Sier :szint :hossz  
  tollatle négyzet :hossz tollatfel  
  Sierpinski :szint :hossz  
vége
```



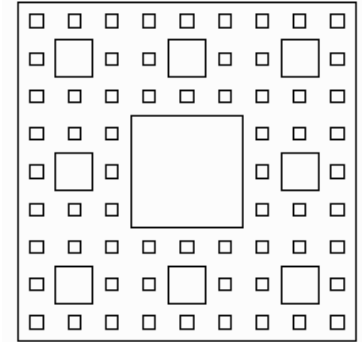


# Logo – Fraktál



## Fraktál – Sierpinski csipke

```
eljárás Sierpinski :szint :hossz
  kivágás :hossz/3
  ha :szint>0
    [ismétlés 4 [ismétlés 2
      [Sierpinski :szint-1 :hossz/3 előre :hossz/3]
      előre :hossz/3 jobbra 90]]
```



```
vége
eljárás kivágás :hossz
  előre :hossz jobbra 90 előre :hossz balra 90
  tollatle négyzet :hossz tollatfel
  jobbra 90 hátra :hossz balra 90 hátra :hossz
vége
```







# Logo – Fraktál



## Fraktál – Sokszögekben más sokszögek

eljárás ábra :db :hossz

ha :db>2

[ismétlés :db

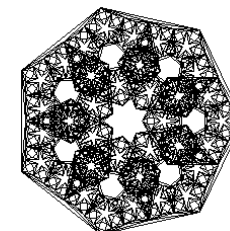
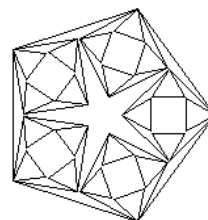
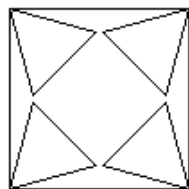
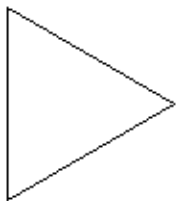
[előre :hossz

jobbra  $360 / :db + (360 / (:db - 1) - 360 / :db) / 2$

ábra :db-1 :hossz/2

balra  $(360 / (:db - 1) - 360 / :db) / 2$  ] ]

vége

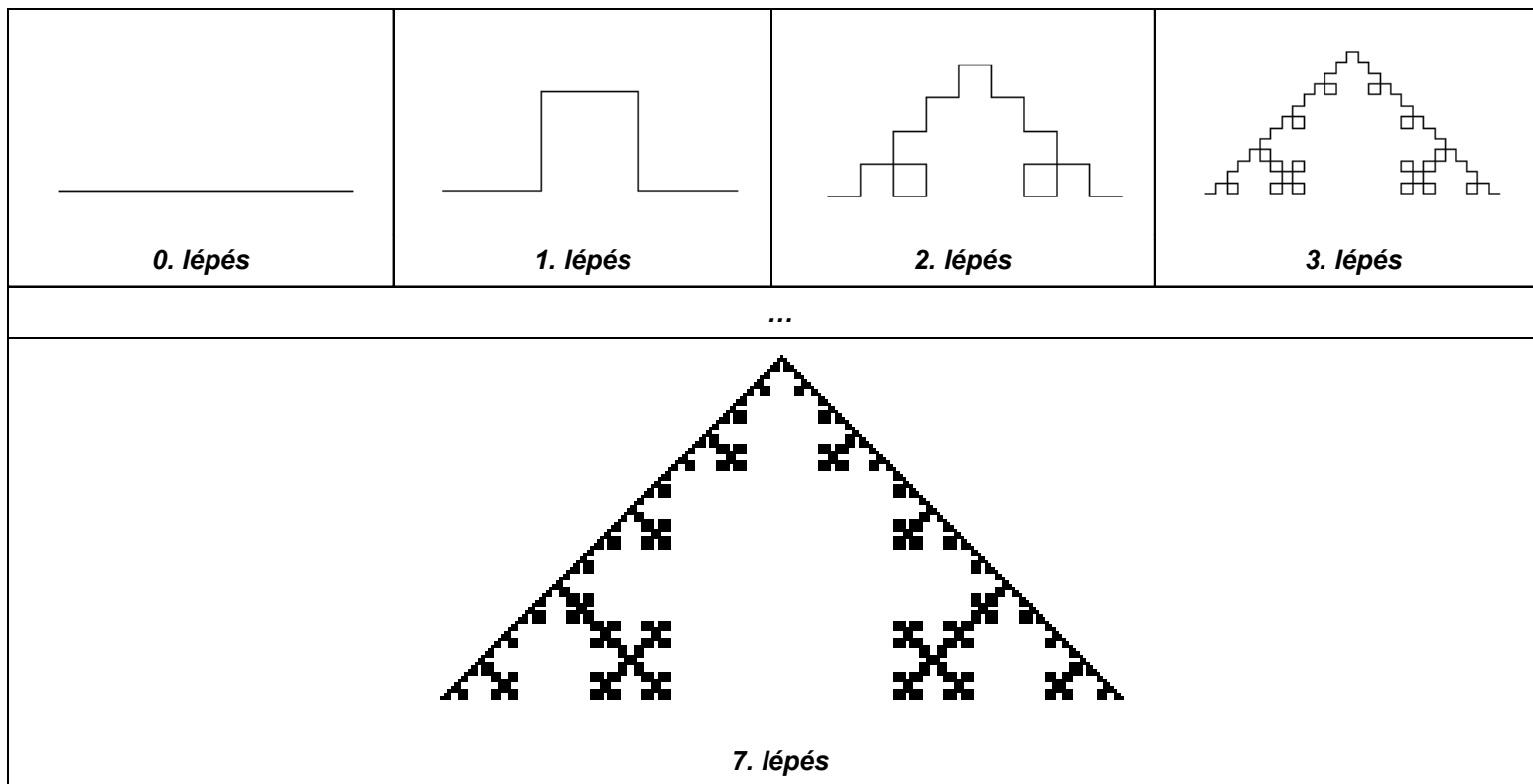




# Logo – Fraktál



## Fraktál

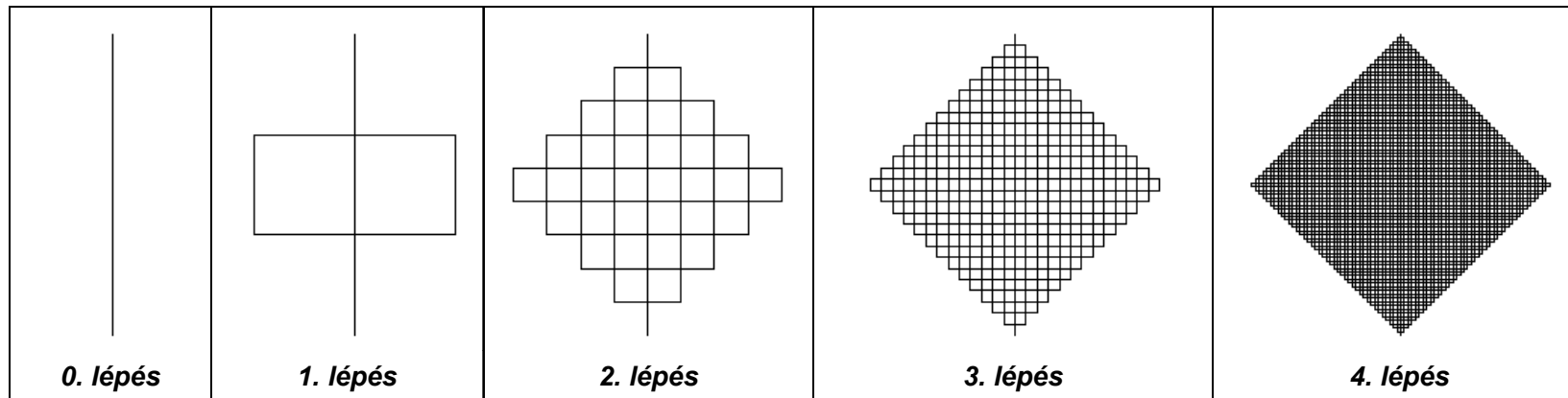




# Logo – Fraktál



## Fraktál

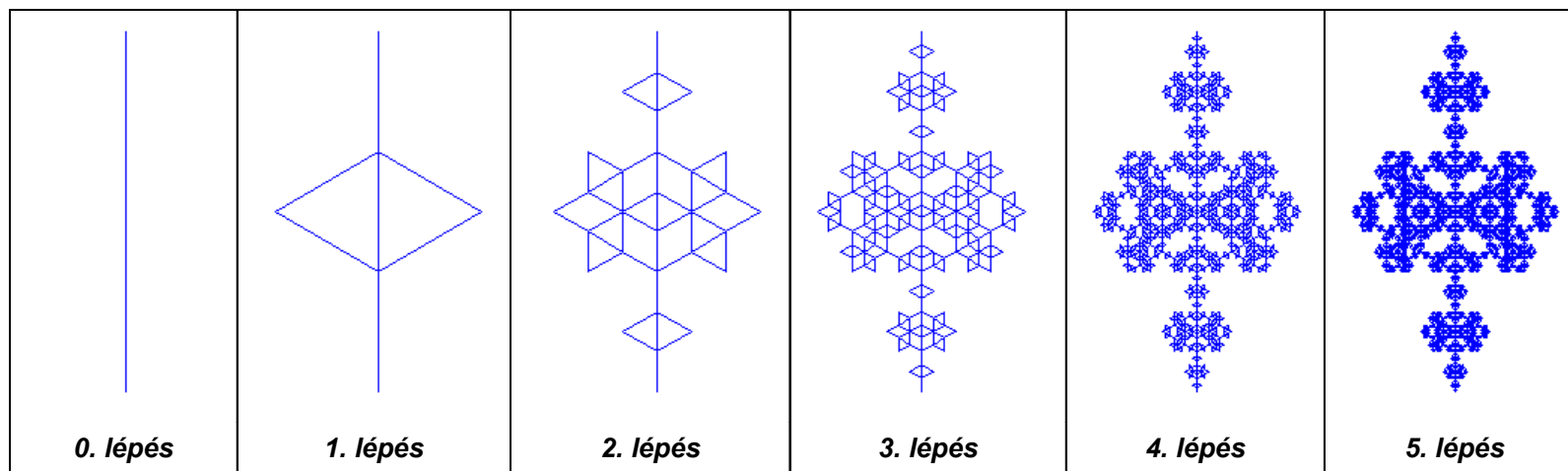




# Logo – Fraktál



## Fraktál

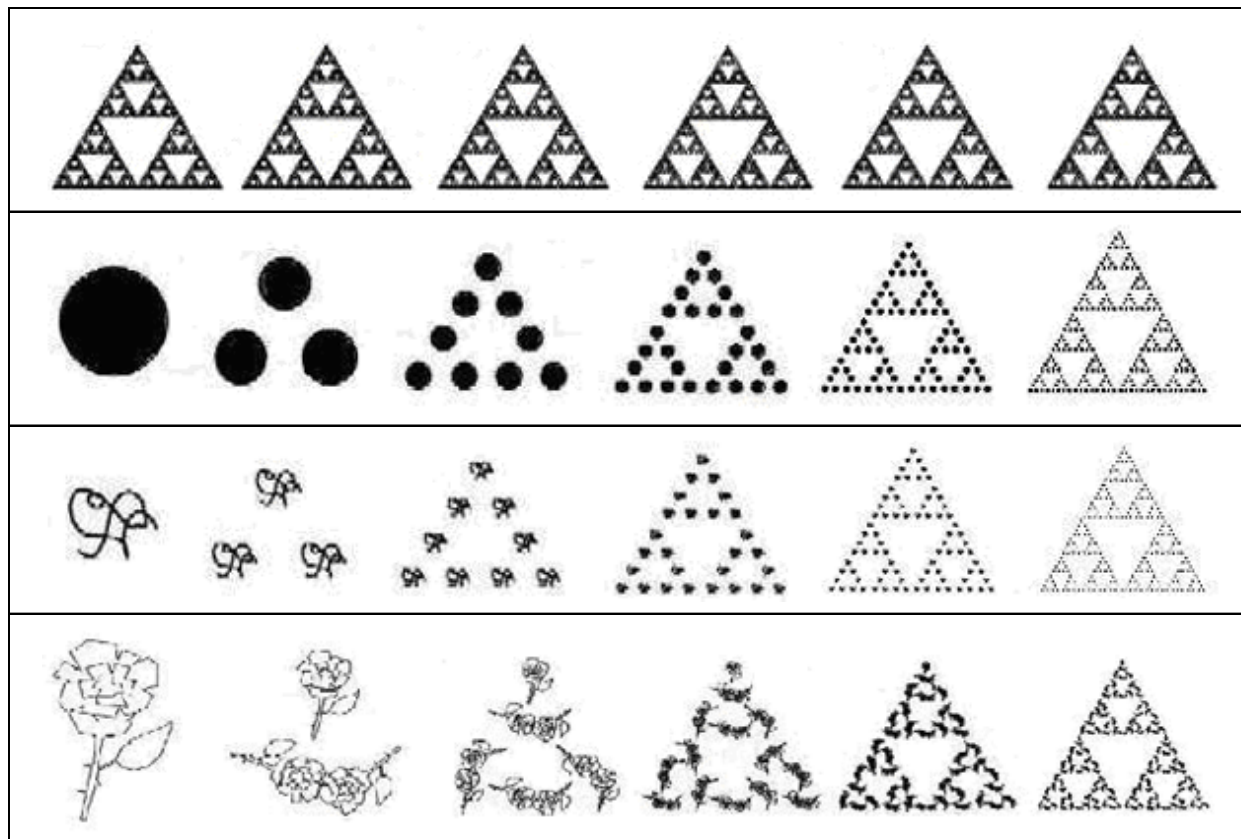




# Logo – Fraktál



## Fraktál





# Logo – vezérelt rajzolás



## Óra

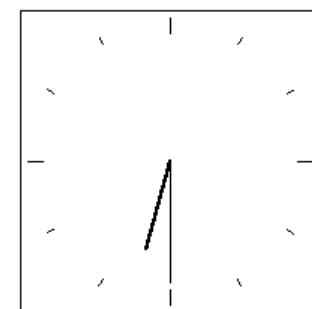
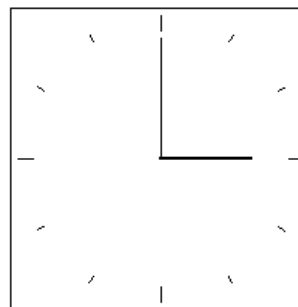
Legegyszerűbb vezérlés a paramétereken keresztül történik. Ilyen feladat lehet az óra rajzolása.

```
eljárás óra :o :p
```

```
  óralap 80 mutató 60 1 6*:p mutató 40 2 (60*:o+:p)/2  
vége
```

```
eljárás mutató :h :v :f
```

```
  tollvastagság! :v jobbra :f előre :h hátra :h  
  balra :f tollvastagság! 1  
vége
```





# Logo – vezérelt rajzolás

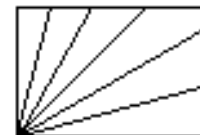
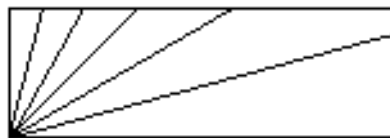


## Rajzolás vonalig

Rajzoljunk egy téglalapba az egyik sarkából 15 fokként szakaszokat! Derékszögű háromszögek átfogóit kell rajzolni, azaz számolható lenne mindegyik méret, de a háromszögek állása a téglalap méretétől függ.

A felhasználandó új művelet – a teknőc egyik érzékelője, a teknőc alatt levő pont színe – pontszín.

A szakaszok addig tartanak, amíg nem érünk festett pontig.





# Logo – vezérelt rajzolás



## Rajzolás vonalig

eljárás téгла :a :b

ismétlés 2 [előre :a jobbra 90

előre :b jobbra 90]

ismétlés 5 [jobbra 15 vonalig xpoz ypoz irány]

vége

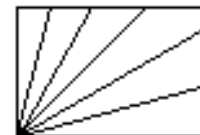
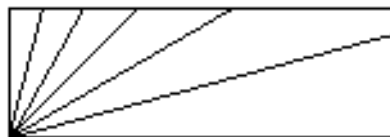
eljárás vonalig :x :y :i

tollatfel **amíg** [pontszín=tollszín] [előre 1]

**amíg** ["fehér = pontszín][előre 1]

tollatle xpoz! :x :y irány! :i

vége







# Logo – vezérelt rajzolás



## Rajzolás vonalig

Ugyanez rekurzívan:

```
eljárás vonalig:x :y :i
```

```
  tollatfel amíg [pontszín=tollszín] [előre 1]
```

```
  vonalig2
```

```
  xypoz! :x :y irány! :i
```

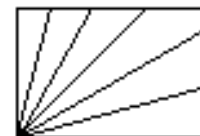
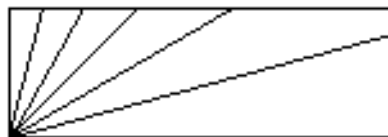
```
vége
```

```
eljárás vonalig2
```

```
  hak 15=pontszín [előre 1 vonalig2 hátra 1]
```

```
  [tollatle]
```

```
vége
```





# Logo – véletlenszerű rajzolás



## Véletlen fa:

Véletlentől függ, hogy balra, illetve jobbra nő-e ág.

eljárás fa :n :h

előre :h

```
ha :n>1 [balra 60 vfa 80 :n-1 :h*2/3 jobbra 120
        vfa 90 :n-1 :h*2/3 balra 60]
```

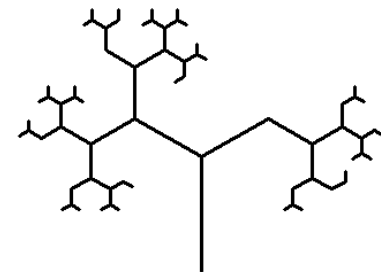
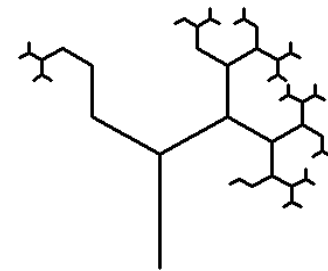
hátra :h

vége

eljárás vfa :s :n :h

```
ha véletlenszám 100<:s [fa :n :h]
```

vége





# Logo – Fraktál



## Fraktál – véletlen Koch görbe

A Koch görbe véletlenített fordulatokkal

```
eljárás Koch :szint :hossz
```

```
  hak :szint = 1 [előre :hossz]
```

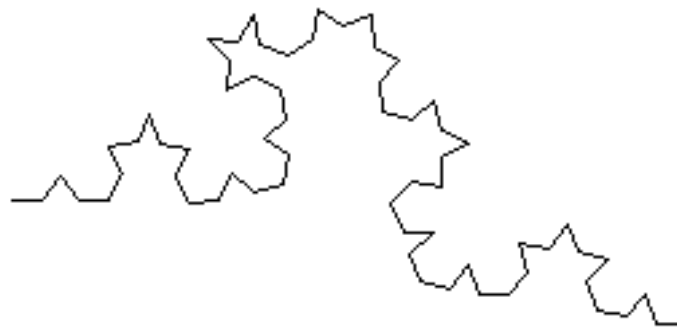
```
[Koch :szint-1 :hossz/3 balra 50+véletlenszám 20
```

```
 Koch :szint-1 :hossz/3 jobbra 100+véletlenszám 40
```

```
 Koch :szint-1 :hossz/3 balra 50+véletlenszám 20
```

```
 Koch :szint-1 :hossz/3]
```

vége





# Programozás nyelvek

## 4. előadás vége